

## Description

# Distributed Dynamic Process Control System

### BACKGROUND OF INVENTION

[0001] To provide users with the tools to implement the large software systems capable of evolving to support ever-changing business processes, many software vendors (for example: <http://www.bpmi.org/members.esp>) introduced what are widely known as Business Process Management (BPM) platforms.

[0002] A typical BPM solution separates the Process and the Activities that together comprise the total business process. Activities are the software modules that perform certain functions without explicit knowledge of each other. A business process, as defined by the Workflow standard—Terminology & glossary, Technical Report WFMC-TC-1011, Workflow Management Coalition, June 1996. Versions 2.0, is simply a set of one or more linked activities that collectively realize a business objective or a pol-

icy goal.

[0003] The Process is represented by the Process Schema or Process Definition that describes the order and conditions in which the Activities should perform to reach the desired results.

[0004] The Schema is executed by a special component often called a Process Server, Workflow Server, Conductor, etc. The functionality of the server includes managing the flow of the process between the activities.

[0005] This approach has severe limitations: all instances of a specific business processes follow the same Process Schema and this approach does not allow an individual instance to deviate from the standard Schema. Nevertheless, some business processes require exactly this type of behavior.

#### **DETAILED DESCRIPTION OF PREFERRED EMBODIMENT**

[0006] Figure 1 illustrates how the present invention combines the Process Schema 101 as the symbolic description of what should be done to complete the business process, under what condition, and by what participants of the process, with the data 102 specific for the individual instance of the process. The result of the combination is the Process Ticket 103.

- [0007] The Process Schema could be expressed in XML formatted structures such as BPEL4WS (<http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>), BPMN (<http://www.bpmi.org/specifications.esp>), ebXML (<http://www.ebxml.org/>), or any other data structure serving the same purpose.
- [0008] The Process Data belongs to the specific instance of the Process. The data may describe a specific Customer Order, an Insurance Claim, an Employee Status Change, an RFP (Request for Proposals) document or the data necessary to support any other business process or operation.
- [0009] The data format depends on the nature of the data. It could be formatted as XML structures, or a Microsoft Word document, a bitmap or vector image file, an Electronic Data Interchange (EDI) document or other common or custom data format appropriate for the specific business process.
- [0010] Figure 2 illustrates the distributed system architecture. The system has one or more Process Controllers 201 (hereafter Controllers) that sends and receives the Process Tickets 202 (hereafter Tickets).
- [0011] The Process Controller may be implemented as a separate

server process, or as a part of another process such as an application server, or as a part of the Execution Agent 203, or as a part of the Execution System.

[0012] Execution System 204 is the software that actually performs the activity. It could be the general-purpose applications such as Microsoft Word or Microsoft Excel, specialized software systems such as ERP or CRM modules, or the software specifically developed for the process control system.

[0013] Controllers 201 communicate with other Controllers, Execution Agents 203, Execution Systems 204 via standard protocols such as SOAP, XML over HTTP, POP, SMTP, etc., as well as other standard and proprietary binary protocols.

[0014] The Controllers pass the Tickets to other Controllers, Agents and Execution Systems for execution and receives it back after the execution.

[0015] Agents act as the intermediary between the Controllers and the Execution System in those cases where the Execution System does not have an interface for the Controller. This may be the case with the third-party Execution Systems.

[0016] For example, if the Execution System is Microsoft Word, the agent is implemented as a plug-in component that

converts the process data into Microsoft Word format (if necessary) and uses Microsoft Word to visualize it. If the data already exists in the Microsoft Word format, the Agent just extracts the data from the Ticket and passes it to the application.

[0017] In addition to interfacing the Execution System to the controller, the Agent also provides Common Process Services (CPS) to the Execution System. The Execution System may request process specific data such as: the participant of the process in the form of Users and Roles, system deadlines, the catalogs of other Execution Systems, etc.

[0018] Figure 3 illustrates the functionality of the Agent. In the most common case, the Agent:–301: Registers with the Controller for specific type of tickets;–302: Receives the Ticket from the controller;–303: Confirms the delivery of the ticket;–304: Converts the process data into the form suitable for the Execution System;–305: Passes the data to the Execution System;–306: Provides the Execution System with the common process services;–307: Provides the programming and user interface for modifications of the Process Schema;–308: Validates and authorizes the changes in the Schema;–309: Receives the completion report from the Execution System;–310: Sends the execu–

tion report along with the updated Ticket to the Controller.

[0019] During the execution of the ticket, the Schema may be changed either by the program logic or by a human user using a user interface provided by the Agent. Schema changes can include the addition and removal of activities and their attributes. The general attributes of the activities may include (but not limited to):

[0020] –The data to be passed to the activity;

[0021] –The execution system;

[0022] –Deadline;

[0023] –User or Role to be assigned to the activity;

[0024] The editing-related attributes of the activity define how much the activity can be changed during the process" execution. Among these attributes are:

[0025] –Ability for the given user to change the Schema

[0026] –Anchor activity that can not be removed from the Schema

[0027] –Data can not be changed

[0028] –Deadline can not be changed

[0029] –Assignee can not be changed

- [0030] Any changes in Schema are applied to the Master Process Schema. The Master Schema is the schema of the process prepared in advance and stored in the system. When the new instance of the process starts, the Master Schema is included in the Ticket.
- [0031] The Master Schema may also be left incomplete. This forces the user of an activity to make changes in the related process instance in order for the Controller to determine the next step in the process execution.
- [0032] The condition of incompleteness of the process schema is detected as an absence of the next step from any activity except the last one.
- [0033] A process schema may have more than one condition of incompleteness.
- [0034] A user of the activity may eliminate one or more conditions of incompleteness by explicitly defining the next step or steps. In this case the process continues until Controller detects the next condition of incompleteness.
- [0035] Figure 4 illustrates the lifecycle of the process schema. It starts 401 as the Master Process Schema. An activity 402 makes the change in the schema. The modified schema 403 defines the next steps in the process. The following activity 404 makes more changes in the Schema.